

COMMUNICATION CONTROL EQUIPMENT

Patent Number: JP63178666
Publication date: 1988-07-22
Inventor(s): YAMADA TOSHIAKI
Applicant(s): RICOH CO LTD
Requested Patent: ☐ JP63178666
Application Number: JP19870009079 19870120
Priority Number(s):
IPC Classification: H04N1/32; B41J13/00; G06F3/12; G06K15/00
EC Classification:
Equivalents:

Abstract

PURPOSE: To use proper paper by detecting a print zone based on a facsimile reception data, detecting the print zone based on the result of detection and providing a function discriminating the paper size based on the result of detection.

CONSTITUTION: A communication procedure control section 13 gives a received compression code to a decode line check section 14, which stores it sequentially to a FAX picture information storage section 15 and gives it to a print dot zone detection section 16. The detection section 16 detects from what order bit till what bit order a black data exists at each line based on the compression code, paper size is discriminated based on the result of detection and stores the paper size information to a FAX picture information storage section 15. When a print command is given to the storage section 15 in this state, after the storage section 15 displays the said paper size information on a paper size display section 18, the stored compression code is sent to a decode processing section 19.

Data supplied from the **esp@cenet** database - 12

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑬ 公開特許公報(A)

昭63-178666

⑭ Int.Cl.*

識別記号

庁内整理番号

⑮ 公開 昭和63年(1988)7月22日

H 04 N 1/32
B 41 J 13/00
G 06 F 3/12
G 06 K 15/00

Z-6940-5C
2107-2C
M-7208-5B
7208-5B

審査請求 未請求 発明の数 1 (全19頁)

⑯ 発明の名称 通信制御装置

⑰ 特 願 昭62-9079

⑱ 出 願 昭62(1987)1月20日

⑲ 発 明 者 山 田 俊 明 東京都大田区中馬込1丁目3番6号 株式会社リコー内

⑳ 出 願 人 株 式 会 社 リ コ ー 東京都大田区中馬込1丁目3番6号

㉑ 代 理 人 弁 理 士 大 澤 敬

明 細 書

1. 発明の名称

通信制御装置

2. 特許請求の範囲

1. フアクシミリ装置からのデータを受信してプリンタに出力する通信制御装置において、受信データに基づいて印刷領域を検出する検出手段と、該検出手段の検出結果に基づいて前記プリンタで使用する用紙サイズを判定する判定手段とを備えたことを特徴とする通信制御装置。

3. 発明の詳細な説明

技術分野

この発明は、フアクシミリ装置からのデータを受信してプリンタに出力する通信制御装置に関する。

従来技術

一般に、文書作成装置(ワードプロセッサ)やパーソナルコンピュータ等の情報処理装置においては情報通信の要求が高まっており、このような要求に応えるものとして例えば情報処理装置のプ

リンタに接続してフアクシミリ送信機能を有する通信端末装置(これを「フアクシミリ装置」と称する)からのデータを受信して、この受信したデータをプリンタに印字させる通信制御装置が考えられる。

ところが、このようにフアクシミリ装置からの受信データを情報処理装置のプリンタで印字する場合、送信側の用紙サイズ以上の用紙を使用しなければデータが欠落してしまうという問題を生じる。

目 的

この発明は上記の点に鑑みてなされたものであり、適切な用紙を使用できるようにすることを目的とする。

発 成

この発明は上記の目的を達成するため、受信データに基づいて印刷領域を検出し、この検出結果に基づいて用紙サイズを判定する機能を備えたものである。

以下、この発明の一実施例に基づいて具体的に

特開昭63-178666 (2)

説明する。

第2図はこの発明を実施した通信制御装置を備えた情報処理システムの一例を示すブロック図である。

この情報処理システムは、文書作成編集装置（ワードプロセッサ）あるいはパーソナルコンピュータ等のホスト1と、印刷装置としての例えばサーマルプリンタ、インクジェットプリンタ、レーザプリンタ等のプリンタ2とを備え、ホスト1にはセントロニクスインタフェースコネクタ4、5及びケーブル8を介してファクシミリ受信機を有する通信端末装置（以下では「ファクシミリ装置」と称する）への送信機能及びファクシミリ装置からの受信機能を有する通信制御装置を接続し、この通信制御装置にはセントロニクスインタフェースコネクタ7、8及びケーブル8を介してプリンタ2を接続し、更に通信制御装置は図線10に接続している。

この情報処理システムにおいて、通信制御装置を使用しないで単に情報処理装置として使用する

ときには、ホスト1とプリンタ2とを直接コネクタ4、5及びケーブル8を介して接続すればよい。

このように通信制御装置はホスト1とプリンタ2との間に接続又は取外することができるので、ホスト1及びプリンタ2に何等の変更を伴うことなく、情報処理システムにファクシミリ装置への送信機能及びファクシミリ装置からの受信機能を持たせた通信端末装置として使用することができる。

なお、ファクシミリ装置からの受信のみを行なう場合にはホスト1は不要であり、通信制御装置とプリンタ2とを接続することによってファクシミリ装置からの受信機能を有する通信端末装置として使用できる。

次に、この通信制御装置の受信機能の概要を第1図を参照して説明する。

相手先ファクシミリ装置から回線制御装置（NCU）11及びモデム12を介してデータが送信されてきたときには、送信手順制御部13は相手

先ファクシミリ装置から受信した圧縮コード（FAX圧縮情報）をデコードラインチェック部14に送す。そこで、このデコードラインチェック部14は、圧縮コードの長さランレングスチェックをして正しいラインか否かを判定し、正しいラインのときには受信した圧縮コードをFAX圧縮情報蓄積部15に順次格納し、また印刷ドット領域検出部16は受信した圧縮コードを与える。

この印刷ドット領域検出部16は受信した圧縮コードに基づいて各ライン毎に何ビット目から何ビット目まで黒データがあるかを検出し、この検出結果に基づいて各頁毎に対応する用紙サイズを判定し、この判定した用紙サイズを示す情報（用紙サイズ情報）をFAX圧縮情報蓄積部15に格納する。

この状態で、受信印刷指示部17によつてFAX圧縮情報蓄積部15に対して受信データの印刷指示が与えられると、FAX圧縮情報蓄積部15は印刷ドット領域検出部16から受領している用紙サイズ情報を紙サイズ表示部18に表示した後、格

納されている圧縮コード（受信データ）をデコード処理部19に送る。

このデコード処理部19は例えば既知のNSFフレームを使用して相手先通信端末装置が自己と同等の機能を有する通信端末装置（ここでは「CCFAX」と称する）か通常のファクシミリ装置（FAX）かを検知して、この検知結果に応じて相手側がCCFAXのときには受信した圧縮コードをデコード（伸長）すると共にドット密度変換（間引き）してイメージデータを作成し、また相手側がFAXのときには受信した圧縮コードをデコードしてイメージデータを作成し、作成したイメージデータをプリンタ出力データ制御部20へ送出する。

このプリンタ出力データ制御部20はデコード処理部19から受領したイメージデータをプリンタ出力インタフェース（I/F）21を介してプリンタに出力して印刷させる。

第3図はこの通信制御装置の構成を具体的に示すブロック図である。

特開昭63-178666(8)

この通信制御装置は、CPU、ROM、RAM及びI/O等からなるマイクロコンピュータ(CPU)31と、ビット・マップ変換処理、データ圧縮処理、デコードラインチェンジ処理、印刷領域検出及び用紙サイズ判定処理、デコード処理等この装置全体の制御に係わる制御プログラム等を格納したROM32と、ホスト側からのデータ(これを「入力データ」と称する)を変換したビット・マップを格納するワークメモリ及びビット・マップをデータ圧縮して生成した圧縮コード並びに受信した圧縮コードを格納するFAX面情報格納エリア(圧縮コード格納エリア)並びにその他のデータを格納するデータメモリ等を構成するRAM33とを備えている。

これ等によつて第1図の通信手順制御部13、デコードラインチェンジ部14、FAX面情報格納部15、印刷ドット領域検出部16、デコード処理部19及びプリンタ出力データ制御部20を構成している。

また、この通信制御装置は、文書作成編集装置

40あるいはパーソナルコンピュータ等のホスト1側からのデータを入力するプリンタ入力I/Fとしてのセントロニクス受信部インタフェース(I/F)34と、ホスト1からの入力データをプリンタ2等の印刷装置に出力する第1図のプリンタ出力I/F21としてのセントロニクス送信部I/F35とを備えている。

なお、セントロニクスインタフェースで使用する信号の内のストロブ(STROBE)、データ(DATA1~DATA8)、アクノウリツジ(ACKNLG)及びビジー(BUSY)についてはマイクロコンピュータ31に入力し、その他の信号についてはセントロニクス受信部I/F34から直接セントロニクス送信部I/F35に出力している。

さらに、この通信制御装置は、操作スイッチ及び表示器並びにベルを付設した第1図の受信印刷指示部17及び紙サイズ表示部18を含む操作ユニット36を備え、この操作ユニット36との間でのデータ送受を司るスイッチ・LEDポート

37を備えている。

なお、操作ユニット36には、フアクシミリ装置への送信開始を指示及び受信データの印字開始指示並びに手動受信指示等をするためのスタートキーと、送信モードとしてホスト側からの入力データをプリンタに出力するプリンタモードと入力データをフアクシミリ装置(FAX)に送信するFAXモードと入力データをプリンタに出力しFAXに送信するプリンタ・FAXモードとを選択するモードセレクトキーと、自動受信を指示する自動キーと、送信終了指示及び1ページの途中でFAX面情報格納エリアがフル状態になったときに当該ページまでの送信指示又は当該ページ以降のページの送信指示並びに印字終了指示等をするためのストップキーと、FAX面情報格納エリアに格納されている面情報(圧縮コード)のクリア指示等をするクリアキーとを備えている。

また、この操作ユニット36には、送信モードを表示するためのモード表示器と、各種の状態(FAX面情報格納エリアのフル状態等)を表示

するステータス表示器と、用紙サイズを表示する用紙サイズ表示器及びページ数を表示するページ数表示器とを備えている。

さらにまた、この通信制御装置は、公衆電話回線を介してフアクシミリ装置との間で通信を行なうために、マルチプロトコル・シリアルコントロール(MPSC)38を備えると共に、モデム(MODEM)39及び回線制御装置(A-ANCU)40を接続している。

次に、このように構成したこの実施例の作用について第4図以降をも参照して説明する。

まず、ホスト側で作成した文書等をフアクシミリ装置に送信する送信モードにおける入力データの圧縮コード化について説明する。

最初にホスト側から送られてくるプリンタのヘッドコントロールコマンドとこのコマンドの解析について第4図をも参照して説明する。

ホスト側からプリンタに対して送られるヘッドコントロールコマンドのシンボル及び機能は次のとおりである。

特開昭63-178666 (4)

BLF: 逆改行; 印字用紙を送りする。
 LF: 改行; 印字の開始と改行。
 FF: フォームフィード; 次の開始行まで用紙を送る。
 CR: 復帰; 印字の開始と復帰。
 STP: ストップ; ホームポジションに戻る。
 CAN: キャンセル; バッファをクリアする。
 ESC, %, 1, n1, n2: イメージ転送; イメージ印字をする。
 ESC, %, 3, n1, n2: 右移動; 指定ドット数だけ右移動。
 ESC, %, 4, n1, n2: 左移動; 指定ドット数だけ左移動。
 ESC, +, n1: 逆ピッチ改行; 指定ピッチで逆改行する。
 ESC, -, n1: ピッチ改行; 指定ピッチで改行する。
 ESC, 6: 6 L P I 設定; 6 L P I の改行を設定する。
 ESC, 8: 8 L P I 設定; 8 L P I の改行を設定する。
 ESC, R: 初期設定; 電源投入後の状態にする。
 ESC, V: 排出; 用紙を排出する。
 ESC, 2, n1, n2: 複数改行; 指定された行数分改行する。

このようなヘッドコントロールコマンドがホスト側から送出されたときの受信側でのコマンド解釈処理は、第4図に示すように、ポートに入力されたデータを取込んで内部RAMの予め定められた

アドレスが「CMD」のアドレス（以下「CMD」と称す）に格納する。

そして、このCMDに格納したデータが「ESC」か否かを判別して、「ESC」でなければ、そのコマンドが「LF」（ラインフィード: 改行）、「CR」（キャリッジリターン）、「FF」（フォームフィード: 改頁）、「BLF」（逆改行）、「STP」（ストップ）、「CAN」（キャンセル）のいずれであるかを判断する。

また、「ESC」であれば、次のデータをポートからCMDに取込んで、そのデータが「%」か否かを判別し、「%」でなければそのデータが「+」（逆ピッチ改行）、「-」（ピッチ改行）、「R」（イニシャライズ）、「V」（排出）、「Z」（直線改行）、「6」又は「8」（改行ピッチ変更）かを判断する。

さらに、そのデータが「%」であれば、次のデータをポートからCMDに取込んで、そのデータが「1」（イメージモード）、「3」（右移動）、「4」（左移動）かを判断する。

次に、ホストからプリンタに送出される印字データ（文字データ）について第5図及び第8図を参照して説明する。

この実施例ではプリンタとして文字コードを文字パターンに変換するキャラクタジェネレータを内蔵していないプリンタを使用して、ホスト側は文字データをイメージデータで送出し、プリンタ側は1印字行（1ライン）分のイメージデータを受領して改行系コマンドを受けたときに印字動作を開始するものとする。

ここで、プリンタのヘッドをdot 1 ~ dot 24 の24個の印字素子を列設した構成として、1印字行（行方向の印字範囲）のドット数mを例えば1440ドットとする。

このとき、ホスト側は1ドット列（24ドット）分のイメージデータを8ビット（1バイト）単位で3回に分けて転送して、順次1ライン（1440列）分転送することになる。

すなわち、1ドット列（24ドット）分のイメージデータは、第5図(イ)に示す第一転送データ、

同図(ロ)に示す第二転送データ、同図(ハ)に示す第三転送データの順に送られて、各転送データは第8図(イ)、(ロ)、(ハ)に示す領域に印字され、このデータ転送が1ライン（印字範囲）のドット数m（例えばm=1440ドット）回繰返される。

したがって、1ライン分のイメージデータは、第5図に丸付文字で示すように第0バイト、第1バイト、第2バイト、第3バイト、……第nバイトの順で転送される。

このように、ホスト側は横1ビット縦24ビットの1ドット列分のデータを1バイト単位に分けて縦方向に3回転送し、1ラインのドット数（mドット）回横方向に順次転送する。

ところが、データ伝搬は行方向に1ドットライン毎に例えば8ビット単位で行なわなければならない。つまり例えば第5図に破線で示すように第0バイト、第3バイト、……、第21バイトの間一ビットを合わせて8ビット（1バイト）のデータとしてデータ圧縮を行なわなければならない。

このとき、入力されたイメージデータを例えば

特願昭63-178666 (5)

第7図に示すようにそのまま入力順にワークメモリに格納する。つまり例えばワークエリアの所定のアドレスに第0バイトの第0ビット～第7ビットを格納（図中の「0/0」は「第0バイト/第0ビット」を表わす。その他も同様である）し、次のアドレスに第1バイトの第0ビット～第7ビットを格納することが考えられる。

このようにイメージデータをワークメモリに格納すると、データ圧縮時には例えば第0バイトの第0ビットを読出し、次に読出しアドレスを3アドレス分更新して第3バイトの第0ビットを読出し、というようにして第21バイトの第0ビットまでの8ビット分のデータを3アドレスずつ読出しアドレスを更新しながら読出した後、データ圧縮をすることになる。

勿論このようにしてもよいのであるが、これでは入力データのデータ圧縮処理に時間がかかり、特にプリンタとファクシミリ装置に同時出力（送信）するときには1ライン分の印字データ（イメージデータ）のデータ圧縮が終るまで次ラインの

印字データを受けられないのでプリント出力が遅くなるという不都合が生じてくる。

そこで、この実施例においては、ホストからのイメージデータをワークメモリに格納するときに予めデータ圧縮時に使用する1ドットライン毎の8ビット（1バイト）のデータに変換して格納する（これが「ビット・マップに変換（展開）する」処理である）。

つまり、第8図に示すようにワークメモリに予め第0ドットライン～第23ドットラインまで1ドットライン毎に8ビットずつイメージデータを格納するエリアを割付ける。したがって、実際のメモリ空間は各ドットライン毎の24個のエリアとなる。なお、ここでは便宜上各ドットラインエリアの最初のアドレスを第0アドレス、以下第1、アドレス…と称することにする。

そして、第0バイトのデータを受けたときには、第0ビットを第0ドットラインエリアの第0アドレスの第0ビットに格納（図中の「0/0」は「第0バイト/第0ビット」を表わす。その他

も同様である）し、第1ビットを第1ドットラインエリアの第0アドレスの第0ビットに格納し、以下同様にして第8ビット～第7ビットを第1～第7ドットラインエリアの各第0アドレスの第0ビットにそれぞれ格納する。

また、第1バイトのデータを受けたときには、第0ビットを第8ドットラインエリアの第0アドレスの第0ビットに格納し、以下同様にして第1ビット～第7ビットを第9～第15ドットラインエリアの各第1アドレスの第0ビットにそれぞれ格納する。

さらに、第2バイトのデータについても同様の処理を行ない、次に第3バイトのデータを受けたときには、第0ビットを第0ドットラインエリアの第0アドレスの第1ビットに格納し、第1ビットを第1ドットラインの第1アドレスの第1ビットに格納し、以下同様にして第2ビット～第7ビットを第0～第7ドットラインエリアの各第1アドレスの第1ビットにそれぞれ格納する。

これ等の処理を繰返し実行することによつて、第8図に示すように例えば第0ドットラインエリアの第0アドレスの第0ビット、～第7ビットには、入力データの第0、第3、第6、第9、…第21バイトの各第0ビットが8ビットデータとして格納され、同様に例えば第8ドットラインエリアの第0アドレスの第0ビット、～第7ビットには、入力データの第1、第4、第7、第10、第13、第16、第19、第22バイトの各第0ビットが8ビットデータとして格納される。

したがって、例えば第0ドットラインのデータ圧縮を行なうときにはワークエリアの第0ドットラインエリアの第0アドレスから順次1アドレスずつアドレス更新をしながらそれぞれ8ビット（1バイト）のデータを読出すことによつて、直ちにデータ圧縮を行なうことができ、データ圧縮を高効率で行なうことができる。

このようにしてホストからの入力データをビット・マップに変換（展開）する場合、ファクシミリ装置の幅が1728～2560ドットであるの

特開昭63-178666 (6)

に対して、この実施例で使用しているプリンタの印字幅は1440ドットであるので、データ変換時にラインの始めと終りにマージンをとる必要がある。

したがって、入力データをワークメモリにビット・マップとして展開する場合には、概ね入力データの第xバイト目の第yビットは次式に従う変換によつて出力データの1ドットラインエリアの第Xバイト目の第Yビットに配置すればよい。

$$X = 8Fw(x \bmod 3) + \lfloor x/3 \rfloor + Fwy + M1/8 \\ Y = 7 - \lfloor x/3 \rfloor \bmod 8$$

なお、上式中、Fw：フアクシミリ装置の幅（ドット数）、M1：左側マージン量（ドット数）、mod：前側の値を後側の値で割ったときの余り、すなわち例えば $(x \bmod 3)$ はxを3で割ったときの余りを意味する。

また、第Yビットを求める式中の「7」はデータ圧縮との関係でそのままでは左（MSB）からのビット数になるのでこれを右側（LSB）からのビット数に変換するための数値である。

ワークメモリは全体をDCCR共有エリア、逆改行対応エリア、最新行作業エリアの三つに分割している。なお、DCCR（データ圧縮装置）はこの実施例ではマイクロコンピュータで構成している。

最新行作業エリアは、プリントアウト中の最新行のエリアであり、プリンタ・ヘッドが横方向に移動しないでプリントできる大きさ、すなわちプリンタ・ヘッド×FAX幅の大きさ有し、例えばプリンタ・ヘッドを24ドット、FAX幅を1728ビットとしたときには、 $24 \times 1728 = 41472$ ビット＝5184バイトの大きさを有する。

逆改行対応エリアは、ビット・マップの展開は終了しているが、データ圧縮は未了で逆改行で変更可なエリアであり、逆方向の用紙送りに対応できる大きさ、すなわち逆改行可能ドット数×FAX幅の大きさを有する。逆改行をどの程度区めるかは仕様により、逆改行可能ドット数を24ドット（1印字行方向ドット数）とすれば上述し

つまり、例えば第0ドットラインについて言えば、前述したように第0、第3、第6、第8、第12、第15、第18、第21バイトの第0ビットで1バイトのデータを生成する。

このとき、そのままでは第5図に示すように第0バイトの第0ビットが1バイトデータの第7ビット（MSB）に対応し、第3バイトの第0ビットが第6バイトに対応し、以下同様にして第21バイトの第0ビットが第0ビット（LSB）に対応することになるが、データ圧縮処理との関係でこれを前述した第8図にも示すように第0バイトの第0ビットを第0ビット（LSB）にし、第3バイトの第0ビットを第1ビットにし、以下同様にして第21バイトの第0ビットを第7ビット（MSB）にする必要がある。この変換を第Yビットを求める式中で数値「7」から「 $\lfloor x-3 \rfloor \bmod 8$ 」を減算して行なっている。

次に、入力データを上述したビット・マップに展開するために使用するワークメモリの構成について第9図を参照して説明する。

た最新行作業エリアと同じ大きさであり、逆改行を認めなければこのエリアは0になる。

DCCR共有メモリは、DCCRからのメモリアクセスが可能なエリアであり、最低で最新行作業エリアの大きさを有する。

このワークメモリは、これ等の三つのエリア全体を一つのリングとして使用し、このときそれぞれのエリアの境界は各エリアの先頭を示すポイントによつて判断する。

次に、上述したビット・マップ展開処理の一例について第10図を参照して説明する。

まず、ホスト側からのデータを入力し、そのデータがイメージ開始か否かを判断する。

そして、イメージ開始であれば、イメージデータを入力し、そのイメージデータ（上述したように1バイトのデータ）の黒点（・1・のビット）を検出し、この黒点の基点からの変位量を決定して、決定したビット・マップ上の位置に黒点を配置する。

すなわち、ワークメモリを予めクリアしておく

ことによつてワークメモリのすべてのビットに「0」が格納されているので、8ビットのイメージデータの内の「1」になつているビットを見つけて、そのビットが対応するワークメモリのアドレスのビットがワークメモリの前述した第8図の例で第0ドットラインエリアの第0アドレスの第7ビット（基点）に対していくら変位した位置かを決定して、その決定した位置に「1」（黒）を配置する。

このとき、基点に対する変位量の決定は前述した式に従い、イメージデータが第Yバイトであるときにその第Xビットが「1」（黒点）であれば、ビット・マップ上（ワークメモリ上）の該当ドットラインエリアの第Xバイト目の第Yビットに「1」を配置する。

このような処理を1バイトのイメージデータの各ビットについて繰返し実行して、1バイトのイメージデータのビット・マップ上への黒点の配置（展開）が終了したときには再度ホスト側からの次の1バイトのイメージデータを入力するための

処理に戻る。

これに対して、ホスト側からのデータがイメージ開始でなければ、イメージ終了か否かを判別して、イメージ終了でなければ、改行系コマンドか否かを判別する。

そして、改行系コマンドであれば、次のイメージデータをビット・マップに展開するために、ワークエリアの基点を変更し、新たなワークエリアになる部分をクリアした後、黒点の配置を終了したエリアのビット・マップを圧縮コード（ここではMHコード）に変換する。

例えば第8図の例で現在同図に示す状態でワークエリアを使用しているとすると、最新行作業エリアにビット・マップを展開し、その後改行系コマンドが入力されたときには、DCR共有エリアを次のワークエリア（最新行作業エリア）としてその先頭アドレスに基点を変更してクリアした後、黒点の配置を完了したワークエリア（旧最新行作業エリア）のビット・マップを圧縮コードに変換する。

また、イメージ開始でなくイメージ終了になれば、すべてのワークエリアすなわち最新行作業エリア、送改行対応エリア及びDCR共有エリアのビット・マップを圧縮コードに変換する。

このようにして得られた送信データとしての圧縮コードをFAX面情報格納エリアに格納し、送信指示が与えられたときに指定された相手先に送信する。

次に、ファクシミリ装置（白黒と同等の通信制御装置を含む）からの文書受信時の処理について説明する。

まず、相手先ファクシミリ装置から受信した圧縮コード（MHコード）をライン単位でチェックしてFAX面情報格納エリアに格納するデコードラインチェック処理について第11図を参照して説明する。

このラインチェック処理において、まずEOL（エンド・オブ・ライン）コード（以下単に「EOL」と称す）を検出する処理をした後、1ラインのビット数をカウントするためのラインビット

カウンタLINBCTを「0」にリセット（LINBCT←0）して、ホワイト（白）MHコードを検出するWMH検出処理に移行する。

このWMH検出処理でホワイトMHコードを検出したときには、そのMHコードに対応するランレングスのビット数をラインビットカウンタLINBCTに加算（LINBCT←RUN+LINBCT）した後、ラインビットカウンタLINBCTのカウント値が1ラインの正しいビット数（ここでは「1728」ビットとする）を越えている（LINBCT>1728）か否かをチェックする。

このとき、ラインビットカウンタLINBCTのカウント値が「1728」を越えていなければ、ターミネートコードTCか否かを判別して、ターミネートコードTCでなければすなわちメイクアップコードMCであれば再度WMH検出処理に戻り、ターミネートコードTCであればブラック（黒）MHコードを検出するBMH検出処理に移行する。

このBMH検出処理でブラックMHコードを検出したときには、そのMHコードに対応するランレングスのビット数をラインビットカウンタLINBCTに加算(LINBCT←RUN+LINBCT)した後、ラインビットカウンタLINBCTのカウンタ値が1ラインの正しいビット数「1728」を超えている(LINBCT>1728)かどうかを判別する。

このとき、ラインビットカウンタLINBCTのカウンタ値が「1728」を超えていなければ、ターミネートコードTCかどうかを判別して、ターミネートコードTCでなければすなわちメイクアップコードMCであれば再度BMH検出処理に戻り、ターミネートコードTCであればWMH検出処理に戻る。

これに対して、WMH検出処理又はBMH検出処理でMHコードでないときには、EOLかどうかを判別し、EOLでなければそのラインはエラーラインであるのでエラーラインを示すEOL(ここでは「EOL(1)」で示す)をFAX面情報

密接エリアに格納する。同時に、WMH検出処理又はBMH検出処理でMHコードを検出したときにラインビットカウンタLINBCTのカウンタ値が「1728」を超えた(LINBCT>1728)ときにもそのラインはエラーラインであるのでEOL(1)をFAX面情報密接エリアに格納する。

そして、WMH検出処理又はBMH検出処理でEOLを検出したときには、ラインビットカウンタLINBCTのカウンタ値が「1728」か(LINBCT=1728)かどうかを判別する。

このとき、ラインビットカウンタLINBCTのカウンタ値が「1728」であれば、そのラインは正しいラインであるので、真終りを示すRTC(リターン・トゥ・コントロール)を検出するRTCカウンタRTCCNTを「0」にリセットした後、受信したMHコード及び正しいラインを示すEOL(ここでは「EOL(0)」で示す)をFAX面情報密接エリアに格納する。

これに対して、ラインビットカウンタLINB

CTのカウンタ値が「1728」でなければ、そのカウンタ値が「0」か(LINBCT=0)かどうかを判別する。

このとき、ラインビットカウンタLINBCTのカウンタ値が「0」であれば、そのラインはエラーラインであるので、EOL(1)をFAX面情報密接エリアに格納する。

これに対してラインビットカウンタLINBCTのカウンタ値が「0」でなければ、RTCカウンタRTCCNTをインクリメント(+1)した後、そのカウンタ値が「2」かどうかを判別して、カウンタ値が「2」でなければEOL(0)をFAX面情報密接エリアにセットし、またそのカウンタ値が「2」であれば真終りであるので2個のEOL(0)(これを内面的に「RTC」とする)をFAX面情報密接エリアにセットしてこの処理を終了する。

このような処理を行なうことによつて、FAX面情報密接エリアには例えば図12図に示すように、先頭にEOL(00000000010000

000)が格納され、その次に受信した1ラインのMHコードが格納されてこのラインの終りには正しいラインを示すEOL(0000000010000000)が格納され、更に次の1ラインのMHコードが格納されてこのラインの終りには正しいラインを示すEOLが格納され、次のラインはエラーラインであつたのでエラーラインを示すエラーEOL(00000000011000000)が格納され、以後同様にMHコード、EOL又はエラーEOLが格納され、1頁の終りにはRTC(2個のEOL)が格納される。

なお、MHコードとEOLとの間にはバイトジヤステフアイ用のF111(フィル)BJFが適宜付加される。また、EOLはEOLを容易に見つけられるようにバイトバウンダリに合わせて付加するようにしている。更に、上述の説明から明らかではあるが正しいラインのEOLとエラーEOLとの区別は第2バイト目の第7ビット0。をラインステータスビットとして正しいラインのEOLは「0」にし、エラーラインのEOLは